

Remarks:

Claims 1, 3-6, 8-12, 14 and 16-24 are pending in the present application. Claims 1, 8 and 12 have been amended. Support for the amendments herein can be found, for example, at paragraphs 22 and 33-37 of the applicants' specification. No new matter is believed to be added.

35 U.S.C. §103:

Claims 1, 3-6, 8-12, 14, 17-19 and 21-23 stand rejected under 35 U.S.C. §103(a) as being obvious in view of U.S. Pat. Pub. No. 2002/0129127 to Romero et al. (hereinafter, '*Romero*') in view of U.S. Pat. No. 7,197,547 to Miller et al. (hereinafter, '*Miller*'). According to the MPEP §706.02(j), to establish a *prima facie* case of obviousness, the prior art reference must teach or suggest all the claim limitations<sup>1</sup>.

With regard to claim 1, *as amended herein*, the applicants assert that *Romero* in view of *Miller* fails to teach or suggest at least:

A method of configuring a load balancer for dispatching client requests in a server farm comprising...

...initializing a load balancer by identifying to said load balancer, address information for a plurality of servers in a server farm;

...individually polling each of the plurality of servers by the load balancer so as to obtain a configuration file from each polled server if available, wherein:

each obtained configuration file contains the parameters necessary to configure the load balancer for bringing the associated polled server on-line;

In asserting the Rejection, the Examiner argues that the configuration file 300 of *Romero* reads on the claimed "configuration file". The applicants respectfully traverse this argument. In this regard, claim 1, *as amended herein* clarifies that each server of a plurality of servers is polled for its own configuration file. That is, the claimed load balancer polls each of a plurality of servers to obtain a unique configuration file from each polled server, thus potentially resulting in a plurality of different configuration files available to the load balancer. Moreover, claim 1, *as amended herein*, clarifies that each obtained configuration file contains the parameters necessary to configure the load balancer for bringing the associated polled server on-line.

To the contrary, in *Romero*, an agent 170 is responsible for creating a single configuration file, which is communicated to a load balancer. In particular, a partition manager 150 may determine the configuration of the partitions 160-162 on the server 140 and provide the necessary configuration information to an agent 170. Alternatively, the agent 170 may obtain the configuration information from the server 140 directly, such as by probing or polling the server 140, the partition manager 150, the operating system, individual partitions, etc., to identify the partitions 160-162 and determine the configuration thereof<sup>2</sup>.

FIG. 3

PARTITION PROFILE				
PARTITION ID	NETWORK ADDRESS	CORRESPONDING CONFIGURATION	ASSIGNED RANK	
A	1.1.1.1	RESOURCE ALLOCATION = 20% RESOURCE UTILIZATION = 50% AVG RESPONSE TIME = 0.5 ms	3	
B	1.1.1.2	RESOURCE ALLOCATION = 50% RESOURCE UTILIZATION = 35% AVG RESPONSE TIME = 0.2 ms	1	
C	1.1.1.3	RESOURCE ALLOCATION = 30% RESOURCE UTILIZATION = 40% AVG RESPONSE TIME = 0.3 ms	2	
:	:	CONFIGURATION n		RANK n
PARTITION n	1.1.1.n			

As best seen in Fig. 3 of *Romero*, which is reproduced above, the assembled information gathered by the agent 170 is used by the agent 170 to create a single partition profile 300 for maintaining the configuration of the partitions 160-162 and preferably comprises for each partition, a partition identification 310, a network address 320 (e.g., an IP address, port, etc.), a corresponding configuration 330, and an assigned rank 340<sup>3</sup>.

Moreover, *Romero* is completely silent with regard to, and fails to teach or suggest, that the load balancer obtains configuration files from each of the plurality of servers. As will be described in greater detail above, the only disclosed file in *Romero* is a partition profile that is created by the agent 170 to assemble information for the load balancer. In this regard, the only disclosure in *Romero* is that the agent 170 obtains information by probing or polling the server for metrics such as “Resource Allocation”, “Resource Utilization” and “Average Response

<sup>1</sup> See also, *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991); MPEP § 2143 - § 2143.03

<sup>2</sup> See for example, paragraphs 29 and 31 of *Romero*.

<sup>3</sup> See for example, paragraphs 32 and 33 of *Romero*.

Time".

Thus, in *Romero*, the agent 170 (regardless of where within the system the agent resides) is responsible for obtaining configuration parameters for the partitions/software servers 160-162. The agent then creates/builds a single partition profile 300 based upon the various obtained parameters of all of the servers and communicates the profile to the load balancer<sup>4</sup>.

In *Miller*, there is no teaching or suggestion whatsoever, as to how the load balancing algorithm is derived. Moreover, *Miller* arguably teaches away from the claimed invention in that critical to the invention in *Miller* is the use of a globally accessible database 220 for storing critical operational parameters, i.e., session and/or application state data similar to that stored on each of the plurality of data caches 250<sup>5</sup>. The application and session state data relate to customer sessions initiated for customer handled by each server/farm unit is stored in the database 220<sup>6</sup>. As such, any server can complete a transaction previously handled by another server by retrieving the appropriate session data from the common database.

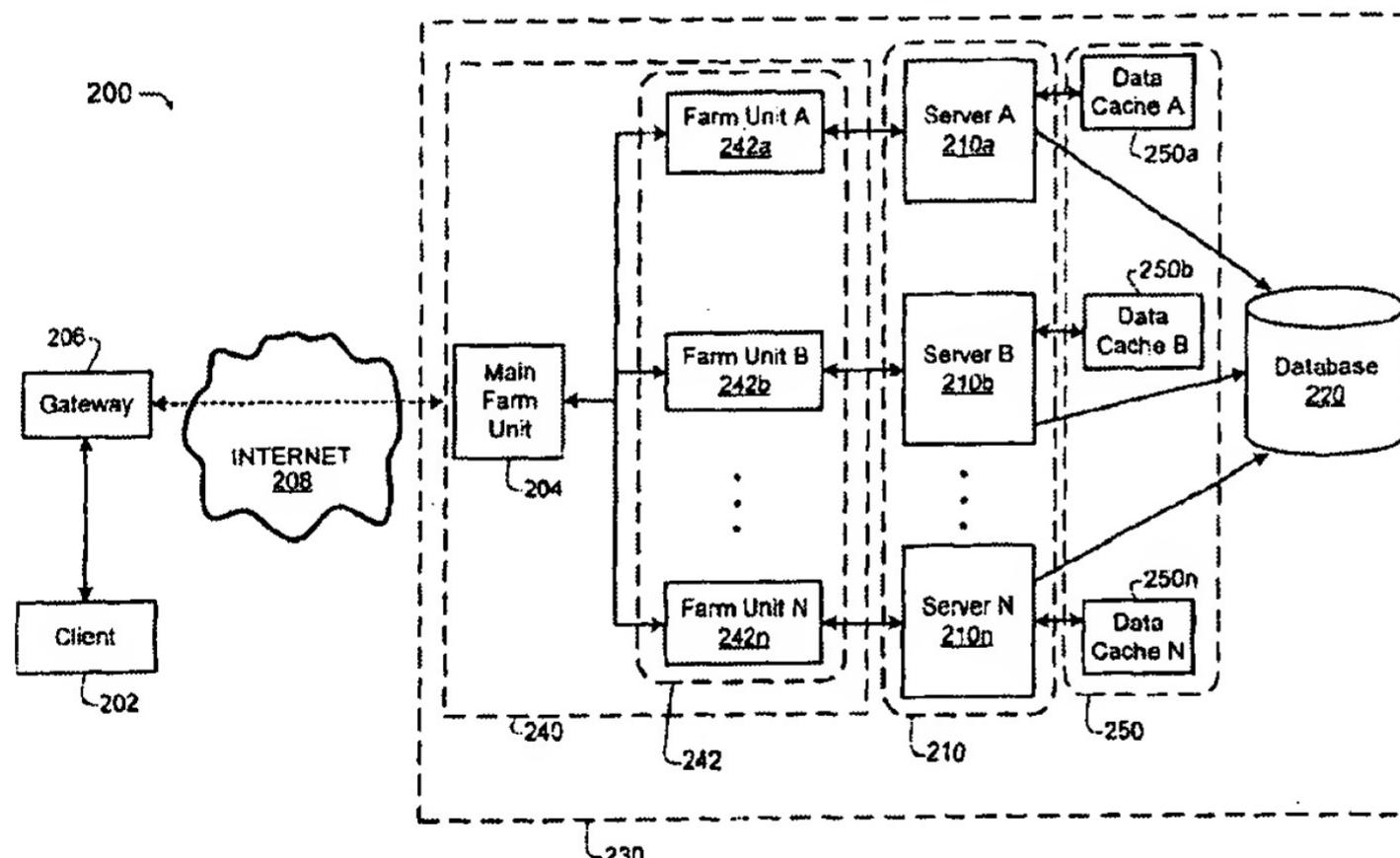


Fig. 2

With regard to claim 1, *as amended herein*, the applicants assert that *Romero* in view of *Miller* further fails to teach or suggest at least:

<sup>4</sup> paragraph 16, 32 and 33 and Fig. 1 of *Romero*.

<sup>5</sup> See for example, *Miller*, Col. 6, lines 3-9.

<sup>6</sup> See for example, *Miller*, Col. 8, lines 4-32.

A method of configuring a load balancer for dispatching client requests in a server farm comprising...

...wherein:

...polling is implemented by repeating for each of said plurality of servers:

selecting by said load balancer, a next one of said plurality of servers;

sending a request across a corresponding network from said load balancer to said next one of said plurality of servers for a corresponding configuration file by specifying a predefined Uniform Resource Locator (URL) using a standard network protocol request, wherein the corresponding configuration file is stored in a local memory of said next one of said plurality of servers identified by the predefined Uniform Resource Locator (URL) and contains parameters including variables to be applied for configuring a load balancing scheme implemented by said load balancer;

waiting for a reply from said next one of said plurality of servers;

receiving either said corresponding configuration file formatted into a markup language supported by the load balancer or an error message from said next one of said plurality of servers...

In this regard, the Examiner further argues that *Romero* teaches that the configuration profile 300 is stored in a local memory of the agent 170. The applicants respectfully point out that claim 1 has been amended herein to clarify that each of a plurality of servers stores its own configuration file in local memory to that server. Moreover, each of the plurality of servers stores its configuration file in such a way that the configuration file is requested by specifying a predefined Uniform Resource Locator (URL) using a standard network protocol request.

The Examiner acknowledges that *Romero* fails to teach or suggest obtaining the configuration file from each of the plurality of servers wherein each configuration file is stored in a local memory of its associated server at a predefined URL<sup>7</sup>.

Correspondingly, the applicants respectfully point out that, as noted above, the configuration file 300 in is not stored in a local memory to any of the various partitions 160, 161, 162<sup>8</sup>. Moreover, there is no teaching or suggestion in *Romero* that the load balancer request the configuration file 300 by specifying a predefined Uniform Resource Locator (URL) using a standard network protocol request.

As such, the Examiner relies upon the teaching in *Miller* with regard to processing “test requests”<sup>9</sup>. However, the applicants respectfully traverse the Examiner’s application of *Miller* to claim 1, *as amended herein*.

In *Miller*, each server is assumed to already be on-line. As such, there is no teaching or suggestion of using the disclosed test-approach for obtaining configuration files from each of the plurality of servers before bringing the servers online. Moreover, the “test URL” is a health monitoring tool configured so as to cause the corresponding server to perform a series of “health” related tasks, e.g., use desired components of the server’s technology to generate a response to the test request. Thus, even though a request is sent to a URL for health status monitoring, the processor at the server being monitored does not retrieve and return a configuration file. Rather, that server creates content for the response by performing a series of tests. This however, fails to teach or suggest obtaining a configuration file from each of the servers wherein each configuration file is stored in a local memory.

As noted in *Miller*, the test URL is a query mechanism that causes the receiving server to perform a series of tests to create content. There is no teaching or suggestion that the content is stored in memory or that the content even relates to configuring a load balancing algorithm. This

---

<sup>7</sup> See the Office action page 3, last paragraph, mailed April 08, 2008.

<sup>8</sup> See for example, paragraph 16 and Fig. 1 of *Romero*.

<sup>9</sup> See the Office action page 4, last paragraph, mailed April 08, 2008.

is further evidenced by the intent of the health monitoring, i.e., to verify the status of an already configured server<sup>10</sup>. Also, *Miller* fails to teach or suggest individually polling each server as recited in claim 1 as amended herein<sup>11</sup>.

With regard to claim 1, *as amended herein*, the applicants assert that *Romero* in view of *Miller* further fails to teach or suggest at least:

A method of configuring a load balancer for dispatching client requests in a server farm comprising...

... validating parameters in said corresponding configuration file if received...

Still further, there is no teaching or suggestion in *Romero* or *Miller* of validating the parameters returned in a response, as recited in claim 1, *as amended herein*. Rather, as best seen in Fig. 6, there is a two part evaluation. If the health monitoring request times out, the corresponding server is disabled. If a response is received and the server is currently disabled, then the server may be re-enabled<sup>12</sup>.

With regard to claim 1, *as amended herein*, the applicants assert that *Romero* in view of *Miller* further fails to teach or suggest at least:

A method of configuring a load balancer for dispatching client requests in a server farm comprising...

A method of configuring a load balancer for dispatching client requests in a server farm comprising...

configuring a load balancing algorithm by said load balancer in accordance with said parameters that were read out of each corresponding configuration file...

*Romero* is silent with regard to, and fails to teach or suggest receiving either a configuration file or an error message in response to requesting a configuration file. Rather, as noted in greater detail above, the agent simply polls the server for desired metrics. Correspondingly, there is no teaching or suggestion anywhere in *Romero* of validating parameters from configuration files associated with each partition. Rather, in *Romero*, the load

---

<sup>10</sup> See for example, *Miller*, Col. 11, lines 29-53.

<sup>11</sup> See for example, *Miller*, Col. 11, lines 15-28.

<sup>12</sup> See for example, *Miller*, Col. 11, lines 29-67.

balancer relies upon the agent 170 to poll for desired information that the agent needs to compute a ranking of partitions and to assemble metrics and other information that identify partitions to be managed by the load balancer.

Still further, *Romero* is silent with regard to, and fails to teach or suggest configuring a load balancer in accordance with the parameters read out from the corresponding configuration files. Rather, in *Romero*, the agent 170 (regardless of its location on the system) consolidates the partition address and configuration information into a single partition profile 300<sup>13</sup> (seen above) that summarizes configuration information for all of the corresponding partitions of the corresponding server(s). The agent also computes a ranking 340 based upon the partition parameters<sup>14</sup>. Thus, the load balancer in *Romero* receives a single profile that identifies each associated server and the necessary load balancing characteristics and pre-computed ranking to perform load balancing functions.

In view of the amendments and clarifying comments herein, the applicants respectfully request that the Examiner withdraw the rejection to claim 1, and the claims that depend there from, under 35 U.S.C. §103.

Independent claims 10 and 12 have been amended herein to recite elements similar to that those in claim 1. As such, the arguments set out above with reference to claim 1 apply by analogy to claims 10 and 12. In view of the amendments and clarifying comments herein, the applicants respectfully request that the Examiner withdraw the rejection to claims 10, 12, and the claims that depend there from, under 35 U.S.C. §103.

35 U.S.C. §103:

Claims 16, 20 and 24 are rejected under 35 U.S.C. §103(a) as being obvious over *Romero* in view of *Miller* and U.S. Pat. No. 6, 286,038 to Reichmeyer (hereinafter, ‘Reichmeyer’).

---

<sup>13</sup> See Fig. 3 of *Romero*.

<sup>14</sup> See paragraph 33 of *Romero*.

According to the MPEP §706.02(j), to establish a *prima facie* case of obviousness, the prior art reference must teach or suggest all the claim limitations<sup>15</sup>.

The applicants' respectfully assert that claims 16, 20 and 24 are patentable at least by virtue of being dependent upon a base independent claim, which applicants believe to be patentable as set out in greater detail herein. In view of the amendments and clarifying comments herein, the applicants respectfully request that the Examiner withdraw the rejection to claim 1, and the claims that depend there from, under 35 U.S.C. §103.

Conclusion:

For all of the above reasons, the applicants respectfully submit that the above claims recite allowable subject matter. The Examiner is encouraged to contact the undersigned to resolve efficiently any formal matters or to discuss any aspects of the application or of this response. Otherwise, early notification of allowable subject matter is respectfully solicited.

Respectfully submitted,  
Stevens & Showalter, L.L.P.

By /Thomas E. Lees/  
Thomas E. Lees Reg. No. 46,867

7019 Corporate Way  
Dayton, Ohio 45459-4238  
Phone: 937-438-6848  
Fax: 937-438-2124

July 1, 2008

---

<sup>15</sup> See also, *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991); MPEP § 2143 - § 2143.03